AIAA Aviation
22-26 June 2015, Dallas, TX
15th AIAA Aviation Technology, Integration, and Operations Conference

Optimization of Push Back Time Windows That Ensure Conflict Free Ramp Area Aircraft Trajectories

William J. Coupe * and Dejan Milutinović †
Jack Baskin School of Engineering, University of California, Santa Cruz, CA 95064, USA
Waqar Malik ‡
University of California, Santa Cruz, NASA Ames Research Center, Moffett Field, CA 94035, USA
Yoon Jung §
NASA Ames Research Center, Moffett Field, CA 94035, USA

The scheduling and execution of ramp area operations is a challenging task. Unlike aircraft maneuvers on taxiways, ramp area aircraft maneuvers are typically not confined to well-defined trajectories. The stochastic nature can force ramp area aircraft to slow down or stop along routes to the taxiway spot to avoid a loss of separation. To address this, we provide a tool for ramp controllers to help meet the assigned taxiway spot times. We formulate a mixed integer linear program model to compute the optimal push back windows that ensure conflict free trajectories. Initiating the push back within the optimal window ensures each departing aircraft can proceed along the route to the taxiway spot without having to slow down or stop for other traffic.

I. Introduction

In order to keep up with the increase of traffic density and reduce congestions on the surface of airports, new techniques are required to improve airport throughput while maintaining safe separation among taxiing aircraft. Since key airports that accommodate a large portion of traffic operate at or close to their maximum capacity, an optimization of runway and taxiway operations is necessary.[1] However, although their operations can be improved by adopting an optimal taxiway schedule, its execution ultimately depends on human controllers who control aircraft maneuvers in both ramp area and taxiways.[2]

Most of the prior research on taxiway scheduling has focused on modeling an airport as a graph, i.e., a connected network, with aircraft travelling along the graph edges. In order to solve the optimization problem on the graph authors have used genetic algorithms,[3,4] Mixed Integer Linear Programs (MILPs),[5] hybrids of these,[6,7] constrained search algorithms,[8] and generalized dynamic programming algorithms.[9] The MILP approach has been used in[10,11,12,13,14,15,16,17] to optimize the routing and scheduling of airport surface traffic. The approach has been applied in[18,19] where an optimization model is formulated for taxi scheduling at Dallas-Fort Worth International Airport (DFW). Similar work[20] has formulated the problem to include uncertainties related to constraint satisfaction while uncertainties in aircraft taxiing has been considered in.[21,22,23] These previous works have addressed uncertainties in the active movement area (runways and taxiways), but do not consider the ramp area. Ramp area aircraft have been incorporated in,[24,25] but the trajectories are considered to be deterministic. This paper attempts to address the integration of uncertain ramp area aircraft trajectories with a state-of-the-art optimal taxiway scheduler. To the best of our knowledge, this research is the first attempt to address the taxiway scheduler problem assuming stochastic ramp area aircraft trajectories.[26]

*AIAA Graduate Student Member, Graduate Student, Computer Engineering Department, UC Santa Cruz.
†Associate Professor, Computer Engineering Department, UC Santa Cruz
‡Research Scientist, University Affiliated Research Center, MS 210-8, Moffett Field, CA 94035.
§AIAA senior member and Aerospace Engineer, NASA Ames Research Center, MS 210-6, Moffett Field, CA 94035.

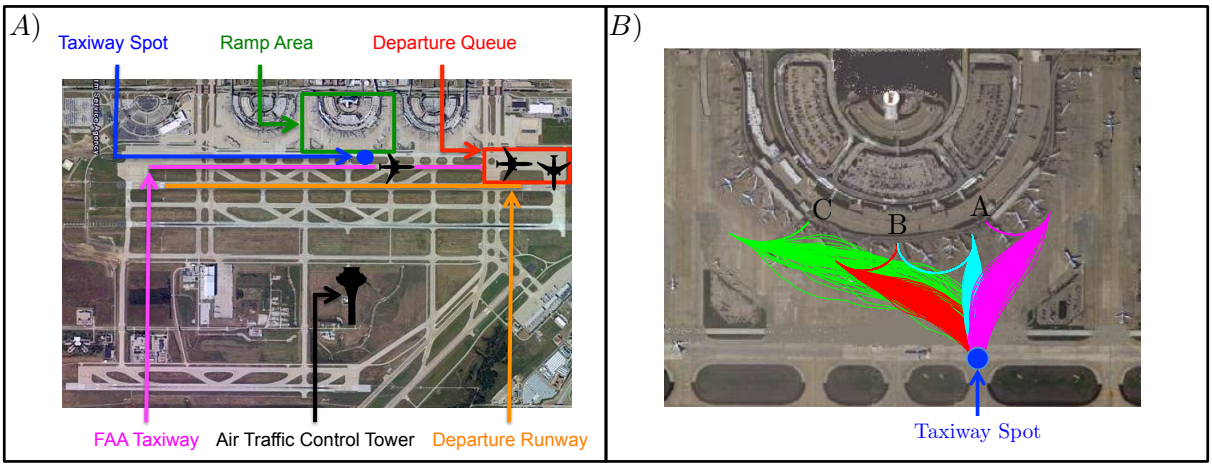

Figure 1. A) Layout of Dallas-Fort Worth International Airport (DFW) with ramp area outlined in green. Departing aircraft push back from their gates and taxi to the departure queue via the taxiway spot. B) Zoomed in view of the green Terminal C ramp area. The departure trajectories from the gate to the taxiway spot (blue) that were sampled from the stochastic model of aircraft trajectories are shown.

The main difficulty in the integration of ramp area aircraft trajectory characteristics into an optimal taxiway scheduling solution is in addressing uncertainties of ramp area trajectories. Unlike aircraft maneuvers on taxiways, ramp area aircraft maneuvers are typically not confined to well-defined trajectories. The shape and timing of the trajectories are subject to uncertainties resulting from pilots decisions as well as other factors involved in ramp area operations, which can impede upon an optimal taxiway schedule plan.

In our previous work[26] we accounted for trajectory uncertainties by developing a stochastic model of aircraft trajectories. The stochastic model was used to generate a probabilistic measure of conflicts within the ramp area, see Fig. 1. The conflict distributions were then used to conservatively schedule conflict free trajectories at the taxiway spot. This method was applied to the DFW Terminal C ramp area to generate optimal schedules for departing aircraft.[26] The method was also applied to the center alley of the Charlotte Douglas International Airport ramp area to generate optimal schedules for both departing and arriving aircraft.[27]

Our previous work has taken a conservative approach to separating aircraft. Using the conservative scheduling approach, we only consider schedules for aircraft i and j that have a zero ratio of conflict. A conflict ratio is estimated by fixing the relative schedule of the two aircraft and computing the ratio of conflicting trajectories to the total number of feasible trajectories. In this paper we build upon our previous work and develop an optimization framework that exploits the structure of the conflict distributions in order to increase the throughput of the optimal schedule.

The throughput of the schedule can be improved upon by considering schedules that have a non-zero ratio of conflicts. For schedules that have a non-zero ratio of conflict, we formulate a MILP that returns the optimal combination of push back sub-windows that ensure conflict free trajectories. We apply the MILP to scheduling departing aircraft within the DFW Terminal C ramp area. We then analyze the increase of throughput that is available when compared to our previous conservative scheduling approach.

This paper is organized as follows. In section II we formulate the optimization problem under consideration. In section III we define the MILP model that we use to solve for the optimal combination of push back sub-windows. Then we provide solutions of the MILP and demonstrate the increase of throughput that can be achieved over the conservative schedule. We then analyze the computational performance of the MILP in comparison to a brute force algorithm that solves for the optimal combination of push back sub-windows. In the last section, we conclude with a discussion of our findings and provide directions for future work.

II. Problem Formulation

The right panel of Fig. 1 shows the DFW Terminal C ramp area. Departing aircraft i begins parked at one of three possible gates labeled with A, B and C. Trajectories that begin from gate B can push

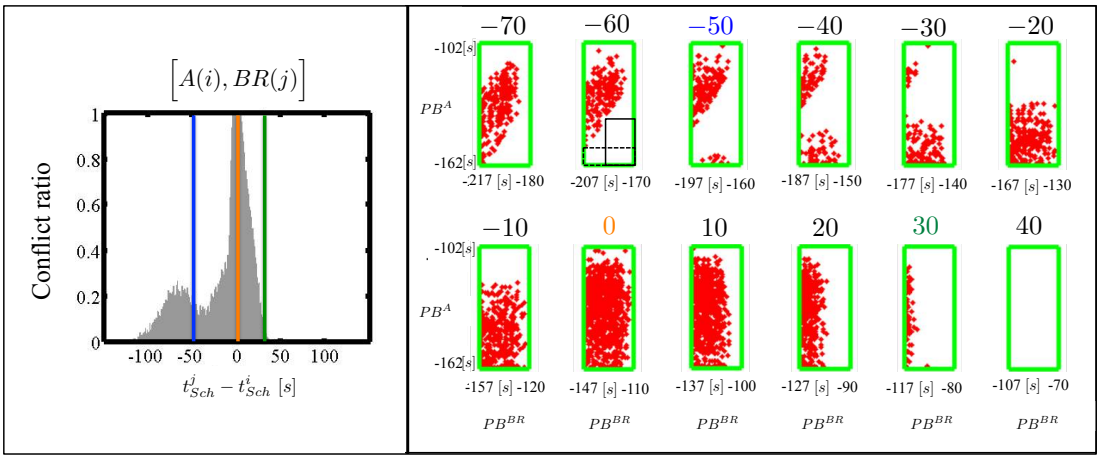

Figure 2. A) Conflict distributions with select cross sections color coded. B) Plot of conflicts between aircraft $A(i)$ and $BR(j)$ for schedules ranging from $t_{BR} - t_A = -70$ to $t_{BR} - t_A = 40$ at a resolution of 10 seconds. For the scheduled difference $t_{BR} - t_A = -60$ two conflict free sub-windows are shown in black solid(dotted) lines.

back with either a left or a right push back maneuver, labeled as BL and BR, respectively. The stochastic model trajectory samples contain spatial and temporal uncertainty and are colored to illustrate the family of possible trajectories. Using the stochastic trajectories we compute a probabilistic measure of conflict among aircraft i and j defined by the difference between their scheduled times at the spot, $t_j - t_i$. The computed conflict distribution is defined by a ratio of the number of pairs of conflicting trajectories to the total number of sampled trajectory pairs. The conflict distribution is estimated by computing the conflict ratio at every whole second, see Fig 2.

For the scheduled spot time differences that have a non-zero ratio of conflicts, we can store and plot the combination of push back times that lead to conflicts. In Fig. 2B the vertical axis represents the push back time of aircraft $A(i)$, PB^A, and the horizontal axis represents the push back time of aircraft $BR(j)$, PB^{BR}. In Fig. 2 we color select cross sections to demonstrate the relationship between the ratio of conflicts (Fig. 2A) defined by the difference between their scheduled spot times and the set of red conflict points (Fig. 2B) defined by the combination of push back times that lead to conflicts for the given difference between their scheduled spot times $t_j - t_i$.

The combinations of push back times that lead to conflicts between aircraft $A(i)$ and $BR(j)$ are plotted (see Fig 2B) in 10s increments for the spot time differences ranging from $t_j - t_i = -70$ to $t_j - t_i = 40$. Given that we are interested in the scheduled spot time difference between two aircraft, we fix the spot time of aircraft $A(i)$ such that $t_A = 0$, and the difference in the scheduled spot time is defined by the spot time of aircraft $BR(j)$. Associated with each difference in scheduled spot time, i.e., $t_{BR} = -70$, is a green rectangle that is defined by the earliest and latest feasible push back times for each aircraft such that the spot time of the schedule is satisfied. Thus, in order to satisfy the spot time $t_A = 0$, aircraft $A(i)$ must push back within the window $PB^A \in [-162, -102]$ and to satisfy the spot time $t_{BR} = -70$ aircraft $BR(j)$ must push back within $PB^{BR} \in [-217, -180]$. For -70 there is a set of combination of push back times that lead to conflicts. These combinations are labeled as red points within the green rectangle (see Fig. 2B).

Consider the distribution of red conflict points for the scheduled spot time difference of -60 seen in Fig. 2 right panel. We observe that in the bottom right of the green rectangle there is a large area that does not contain any red conflict points. If we restrict aircraft $A(i)$ and $B(j)$ to push back within the lower right corner of the green rectangle then we can ensure conflict free trajectories. Two potential solutions are shown where the first solution is shown with a solid black line and the second solution with a dotted black line. Among all possible solutions we would like to find a combination of push back time windows where the minimum time window is maximized. The abstract form of the optimization problem is defined as

$$\max_{t_i^S, t_i^F, t_j^S, t_j^F} \quad J := \min\{t_i^F - t_i^S, t_j^S - t_j^F\} \tag{1}$$

$$\text{subject to:} \quad \forall\, \kappa = (x, y) : \left[x \notin [t_j^S, t_j^F] \vee y \notin [t_i^S, t_i^F]\right] \tag{2}$$

where the cost function J is a function of the four variables t_i^S, t_i^F, t_j^S, t_j^F which represent the earliest and latest push back times for aircraft i and j, respectively. Thus, for aircraft i the variables t_i^S and t_i^F represent the start and finish of the push back window. The four variables together define a combination of push back sub-windows such as the windows labeled with the solid (dotted) black lines. The optimization problem is subject to the constraints that any given conflict point $\kappa = (x, y)$ can not be contained within the optimal combination of push back sub-windows. For any given schedule, at a resolution of 1[s], we consider solving for the optimal combination of push back sub-windows that are constrained to contain no conflicts.

III. Mixed Integer Linear Program (MILP)

Here we provide the mathematical formulation of the abstract optimization problem formulated by expressions (1) and (2). Given any two aircraft i and j, the objective function (1) is used to maximize the minimum time window for both aircraft. While this seems like a reasonable objective function there is a slight problem with this formulation. With this objective function we can not distinguish between two time windows that have equal minimum edge length, as illustrated in Fig. 3A.

In Fig. 3A the minimum edge of the orange combination of time windows and the minimum edge of the blue combination of time windows are equal and defined by t_j^S and t_j^F. Clearly, we would prefer the orange combination of time windows to the blue as aircraft i has a much larger time window to push back within. In order to distinguish between these two solutions we introduce the objective function

$$\max_{t_i^S, t_i^F, t_j^S, t_j^F} \quad J := \left[M + \epsilon(t_i^F - t_i^S + t_j^F - t_j^S)\right] \tag{3}$$

where $M = \min\{t_i^F - t_i^S, t_j^S - t_j^F\}$ is the minimum time window among both aircraft i and j and ϵ is sufficiently small. In the objective function the extra term multiplied by ϵ is added in order to distinguish between two combinations of time windows that have equal minimum edge lengths. Using the objective function (3) with the example depicted in Fig. 3A, we can distinguish between the orange and blue combination of time windows and the orange combination would be selected as optimal. The optimization constraints are described in sequel.

For departing aircraft $i, j \in D$ we introduce the two constraints

$$t_i^F - t_i^S - M \geq 0 \tag{4}$$
$$t_j^F - t_j^S - M \geq 0 \tag{5}$$

that ensure the push back time window for aircraft i and the push back time window for aircraft j are both greater than the minimum time window M. We note that the value M is not a fixed value, but a function of the four variables we solve for.

Similarly, for departing aircraft $i, j \in D$ we introduce the two constraints

$$t_i^F - t_i^S - \delta_{min} \geq 0 \tag{6}$$
$$t_j^F - t_j^S - \delta_{min} \geq 0 \tag{7}$$

that ensure the push back time windows for aircraft i and j are both larger than a predefined value δ_{min}. The value δ_{min} is the minimum acceptable push back window. For example, pilots and ramp area ground crew could find a schedule that requires aircraft to initiate push back within a window of 5 seconds too restrictive to consistently execute. In this paper we use the value $\delta_{min} = 25[s]$ when solving for the optimal sub-windows. The correct value should be determined in conjunction by ramp area controllers and pilots.

For departing aircraft $i, j \in D$ we introduce the four constraints

$$t_i^S - t_i - t_i^{S0} \geq 0 \tag{8}$$
$$t_i^F - t_i - t_i^{F0} \leq 0 \tag{9}$$
$$t_j^S - t_j - t_j^{S0} \geq 0 \tag{10}$$
$$t_j^F - t_j - t_j^{F0} \leq 0 \tag{11}$$

where t_i is the taxiway spot time of aircraft i and t_i^{S0} and t_i^{F0} are the earliest and latest feasible push back times for aircraft i such that the scheduled spot time $t_i = 0$ is enforced. The same definitions apply to the

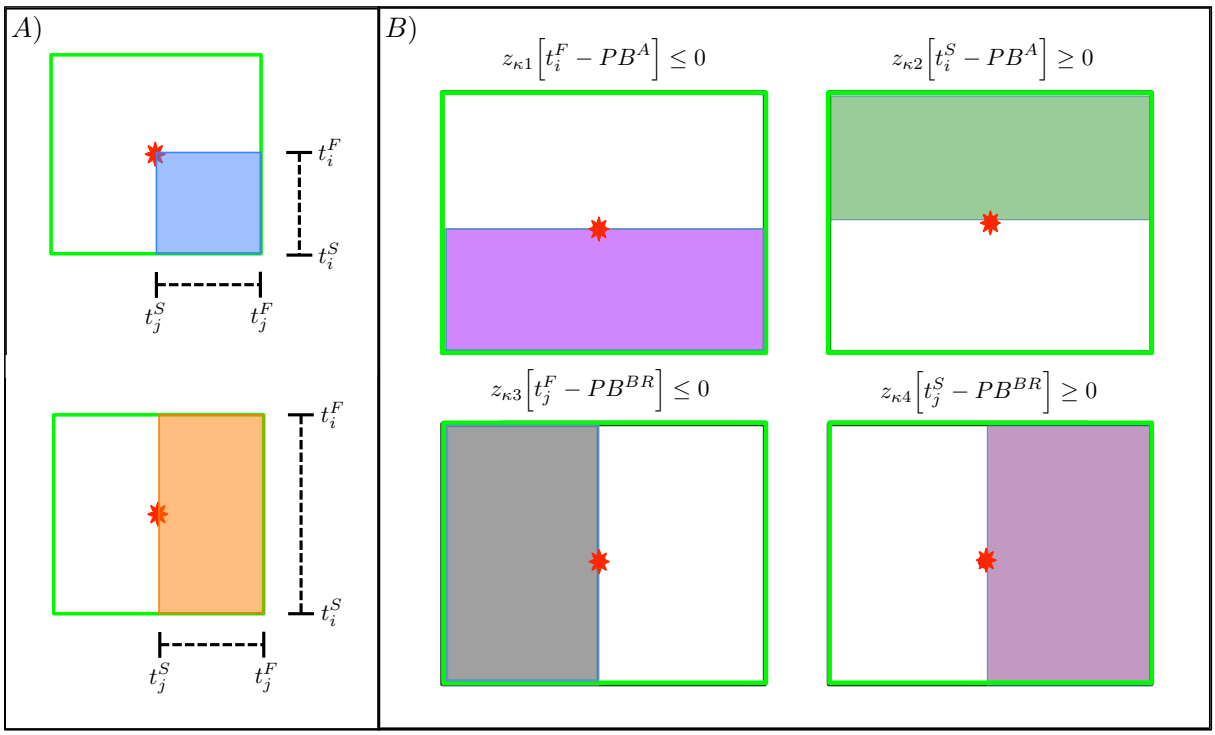

Figure 3. A) The cost function in objective (3) is a function of 4 variables $(t_i^S, t_i^F, t_j^S, t_j^F)$. The minimum edge length of the blue combination of time windows is equal to the minimum edge of the orange combination of time windows. By adding the extra term $\epsilon\Sigma_{i,j} t_{i/j}^F - t_{i/j}^S$ in the cost function we can distinguish between the two rectangles and the orange rectangle is selected as optimal. B) Set of 4 constraints that ensure the optimal combination of push back sub-windows is either above, below, left or right of any single conflict point $\kappa = (PB^{BR}, PB^A)$.

variables for aircraft j. For the scheduled spot time $t_i = 0$, the earliest feasible push back time is defined by $t_i^{S0} = -\max_i(T_i)$ and the latest feasible push back time is defined by $t_i^{F0} = -\min_i(T_i)$. The variable T_i is the trajectory duration of aircraft i that is sampled from the stochastic model. For any given relative schedule, the earliest and latest feasible push back times define the green edges of the rectangle that are seen in Fig. 2. The distribution in trajectory duration is estimated from the robot experiment data which is directly influenced by the human operator.

Constraints (8) - (11) ensure that for any given combination of spot time schedules, given by t_i and t_j, the start and end of the push back sub-windows defined by t^S and t^F must be within the bounds defined by the earliest and latest feasible push back times. This implies satisfying these constraints ensures that there exists a feasible trajectory for aircraft i/j that is capable of meeting the scheduled spot times t_i/t_j without accounting for conflicts. These four constraints provide that the push back windows that we solve for, which are illustrated in black solid (dotted) lines in Fig. 2, are indeed sub-windows of the original green rectangle.

To solve for the optimal push back sub-windows we need information related to the set of conflict points. For example, the conflict points could be used to estimate a distribution that defines the probability of conflict as a function of the combination of push back times. The conflict points could also be used to fit a piecewise linear boundary that separates the level set of conflicting combination of push back times from the level set of conflict-free combination of push back times. In contrast to these approaches, we use the conflict points directly to generate constraints.

The constraints we use are based on an idea that no conflict point should be a convex combination of the optimal combination of time window endpoints. The conflict point $\kappa = (x, y)$ is a convex combination of the optimal time window endpoints of aircraft i if and only if

$$\alpha t_i^F + (1 - \alpha)\, t_i^S = y, \quad \alpha \in [0, 1]$$

This implies that we can enforce that point κ is *not* a convex combination of t_i^S and t_i^F by choosing a value

of α that is either smaller than 0 or greater than 1. For departing aircraft $i, j \in D$, we enforce the following set of seven constraints for each conflict point $\kappa = (PB^{BR}, PB^{A})$.

$$\alpha_{\kappa 1} t_i^F + (1 - \alpha_{\kappa 1})\, t_i^S = PB^A \tag{12}$$

$$z_{\kappa 1}\Big[\alpha_{\kappa 1} - 1\Big] \geq 0 \tag{13}$$

$$z_{\kappa 2}\Big[\alpha_{\kappa 1}\Big] \leq 0 \tag{14}$$

$$\alpha_{\kappa 2} t_j^F + (1 - \alpha_{\kappa 2})\, t_j^S = PB^{BR} \tag{15}$$

$$z_{\kappa 3}\Big[\alpha_{\kappa 2} - 1\Big] \geq 0 \tag{16}$$

$$z_{\kappa 4}\Big[\alpha_{\kappa 2}\Big] \leq 0 \tag{17}$$

$$z_{\kappa 1} + z_{\kappa 2} + z_{\kappa 3} + z_{\kappa 4} = 1 \tag{18}$$

where z_κ is a binary variable. The constraints (12)-(18) ensure that the conflict point $\kappa = (PB^{BR}, PB^{A})$ is not a convex combination of the optimal time window endpoints t_i^S and t_i^F nor a convex combination of the optimal time window endpoints t_j^S and t_j^F. By enforcing this set of constraints we ensure that the conflict point κ is not a convex combination of the optimal combination of push back time window endpoints.

The constraints in equations (12)-(17) can be simplified. From constraints (12)-(14) we can write $\alpha_{\kappa 1}$ as a function of the conflict point κ and the start and end of the push back sub-windows, t_i^S and t_i^F, respectively.

$$\alpha_{\kappa 1}(PB^A, t_i^S, t_i^F) = \frac{PB^A - t_i^S}{t_i^F - t_i^S}$$

Substituting the function $\alpha_{\kappa 1}(PB^A, t_i^S, t_i^F)$ into constraint (13) provides us with the equation

$$\frac{PB^A - t_i^S}{t_i^F - t_i^S} \geq 1$$

which can be enforced by satisfying the inequality

$$PB^A \geq t_i^F$$

Similarly we can substitute the function $\alpha_{\kappa 1}(PB^A, t_i^S, t_i^F)$ into constraint (14) which provides us with the equation

$$\frac{PB^A - t_i^S}{t_i^F - t_i^S} \leq 0$$

which can be enforced by satisfying the inequality

$$PB^A \leq t_i^S$$

Following the same reason, we can transform the three constraints (15)-(17) into the two constraints

$$PB^{BR} \geq t_j^F$$
$$PB^{BR} \leq t_j^S$$

Putting everything together we obtain the following set of five constraints that can be used instead of the seven constraints (12-18) for each conflict point $\kappa = (PB^{BR}, PB^{A})$.

$$z_{\kappa 1}\Big[t_i^F - PB^A\Big] \leq 0 \tag{19}$$

$$z_{\kappa 2}\Big[t_i^S - PB^A\Big] \geq 0 \tag{20}$$

$$z_{\kappa 3}\Big[t_j^F - PB^{BR}\Big] \leq 0 \tag{21}$$

$$z_{\kappa 4}\Big[t_j^S - PB^{BR}\Big] \geq 0 \tag{22}$$

$$z_{\kappa 1} + z_{\kappa 2} + z_{\kappa 3} + z_{\kappa 4} = 1 \tag{23}$$

The set of constraints (19)-(23) enforces that the conflict point κ is *not* a convex combination of the start and end times of the optimal combination of sub-windows defined by $(t_i^S, t_i^F, t_j^S, t_j^F)$. Geometrically speaking, this set of five constraints ensures that any feasible combination of sub-window is either above, below, left or right of the conflict point κ, shown in Fig. 3.

In the optimization problem defined by objective (3) subject to constraints (4)-(11) and (19)-(23), for every conflict point κ we have four nonlinear constraints (19)-(22). These constraints can be linearized as

$$t_i^F - PB^A - (1 - z_{\kappa 1})S \leq 0 \tag{24}$$
$$t_i^S - PB^A + (1 - z_{\kappa 2})S \geq 0 \tag{25}$$
$$t_j^F - PB^{BR} - (1 - z_{\kappa 3})S \leq 0 \tag{26}$$
$$t_j^S - PB^{BR} + (1 - z_{\kappa 4})S \geq 0 \tag{27}$$

where the value of S is sufficiently large. By replacing the nonlinear constraints (19)-(22) with the linear constraints (24)-(27) we obtain a MILP problem that we can pass directly to Gurobi Optimizer[28] to solve. The mixed integer linear program that we pass to Gurobi is defined as

$$\max_{t_i^S, t_i^F, t_j^S, t_j^F} \quad J := \left[M + \epsilon(t_i^F - t_i^S + t_j^F - t_j^S)\right] \tag{28}$$

for aircraft i, j we generate the eight constraints

$$t_i^F - t_i^S - M \geq 0 \tag{29}$$
$$t_j^F - t_j^S - M \geq 0 \tag{30}$$
$$t_i^F - t_i^S - \delta_{min} \geq 0 \tag{31}$$
$$t_j^F - t_j^S - \delta_{min} \geq 0 \tag{32}$$
$$t_i^S - t_i - t_i^{S0} \geq 0 \tag{33}$$
$$t_i^F - t_i - t_i^{F0} \leq 0 \tag{34}$$
$$t_j^S - t_j - t_j^{S0} \geq 0 \tag{35}$$
$$t_j^F - t_j - t_j^{F0} \leq 0 \tag{36}$$

and for each conflict point $\kappa = (PB^j, PB^i)$ we generate the five constraints

$$t_i^F - PB^i - (1 - z_{\kappa 1})S \leq 0 \tag{37}$$
$$t_i^S - PB^i + (1 - z_{\kappa 2})S \geq 0 \tag{38}$$
$$t_j^F - PB^j - (1 - z_{\kappa 3})S \leq 0 \tag{39}$$
$$t_j^S - PB^j + (1 - z_{\kappa 4})S \geq 0 \tag{40}$$
$$z_{\kappa 1} + z_{\kappa 2} + z_{\kappa 3} + z_{\kappa 4} = 1 \tag{41}$$

IV. MILP Optimal Time Window Solutions

Figure 4 illustrates the optimal combination of push back sub-windows for aircraft $A(i)$ and $BR(j)$. Solutions are computed for the differences of taxiway spot times of departing aircraft with a resolution of 1 second. The optimal solution for the scheduled spot time difference $t_j - t_i = 23[s]$ is shown in Fig. 4A and the solution for the scheduled spot time difference $t_j - t_i = -39[s]$ in Fig. 4B. The optimal combination of push back sub-windows are labeled by the purple rectangles, which are by definition within the green rectangles and contain no red conflict points.

The two solutions in Fig. 4 demonstrate a key property of our MILP problem formulation. In Fig. 4A the conflict points appear as a single cloud while the conflict points in Fig. 4B appear to form two disjoint clouds. Our MILP approach addresses the challenge of computing the boundaries around conflict points. Our MILP approach is appealing as the complexity and structure of the clouds of conflict points is not known *a priori*.

Figure 5 provides the minimum time-separation at the taxiway spot between aircraft i and j that ensures conflict free trajectories. In the graph, the directed edge e_{i-j} represents the minimum time separation when

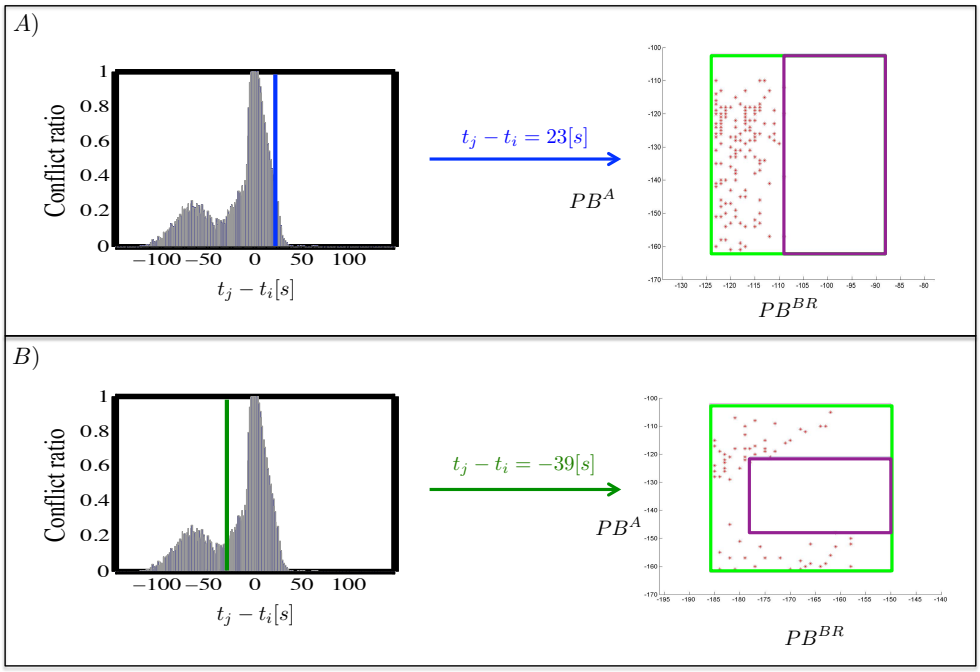

Figure 4. A) Optimal combination of push back sub-windows for the scheduled spot time difference $t_j - t_i = 23[s]$. B) Optimal combination of push back sub-windows for the scheduled spot time difference $t_j - t_i = -39[s]$.

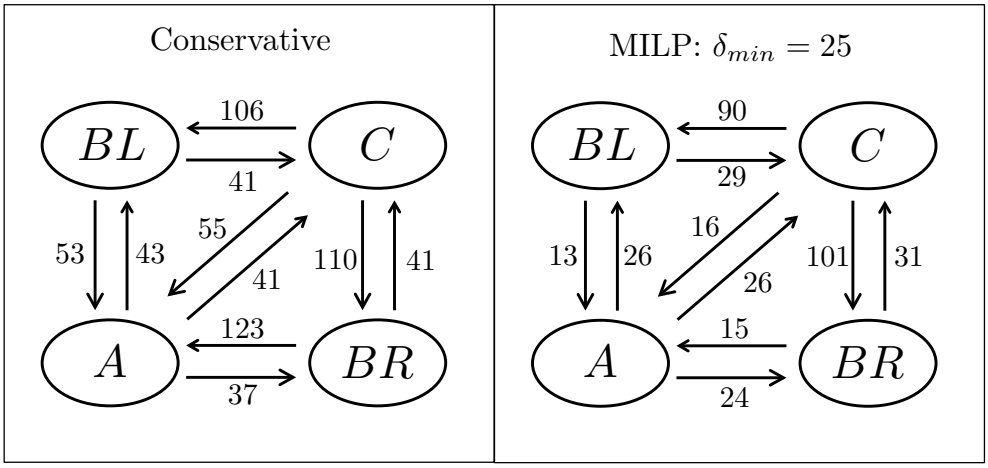

Figure 5. LEFT: Minimum time separation at the taxiway spot using the conservative conflict separation constraints. RIGHT: Minimum time separation at the taxiway spot using the optimal combination of push back sub-windows. Here we assume that the minimum push back time window that we are willing to accept is given by $\delta_{min} = 25$.

scheduling aircraft i followed by j. Figure 5A and 5B provide the minimum time separation for a conservative approach and the MILP approach, respectively. The conservative approach is defined to separate aircraft at the taxiway spot such that there is a zero ratio of conflicts. For the conflict distribution seen in Fig. 2A, for example, we can see that edge $e_{A-BR} = 37$ and $e_{BR-A} = 123$. The MILP formulation exploits the structure of the conflict points and allows us to reduce the minimum time separation between aircraft i and j for all possible sequences.

The minimum-time separation graph in Fig. 5 can be used to schedule aircraft at the taxiway spot.[26] Using a MILP approach, the minimum-time separation can be enforced with constraints, and schedules can be computed to optimize the uninterrupted flow of departure traffic from the gate to the departure queue. Using the conservative graph as constraints, the optimal schedule is computed as $t_A = 0, t_B = 37$ and

Algorithm 1 Brute Force Algorithm

Set OptCost = 0
Set left = $t_j - t_j^{S0}$
Set right = $t_j - t_j^{F0}$
for t_j^S = left:right $-\delta_{min}$ **do**
  for t_j^F = left $+\delta_{min}$: right **do**
    • Solve for t_i^S and t_i^F that provides the largest vertical window that contains no conflict points.
    • cost = $\min[t_i^F - t_i^S, t_j^F - t_j^S] + \epsilon(t_i^F - t_i^S + t_j^F - t_j^S)$
    if cost > OptCost **then**
      • OptCost = cost
      • OptWindows = $(t_i^S, t_i^F, t_j^S, t_j^F)$
    end if
  end for
end for
Return OptWindows

A)

Number of Points in Domain

Area of Domain	MILP	100	200	300	400	500
	10,000	0.60	1.82	4.35	9.35	14.77
	40,000	0.68	2.36	5.75	9.28	18.46
	90,000	0.73	2.73	7.22	12.29	15.90
	160,000	0.70	2.47	7.01	15.25	18.39
	250,000	0.77	2.66	10.27	17.98	20.26

Area of Domain	BF	100	200	300	400	500
	10,000	1.29	3.71	5.46	6.23	8.70
	40,000	4.86	10.74	26.64	32.57	39.20
	90,000	20.54	41.21	35.97	60.56	93.88
	160,000	38.49	51.95	101.40	145.79	171.50
	250,000	53.99	112.91	168.21	190.42	266.95

B)

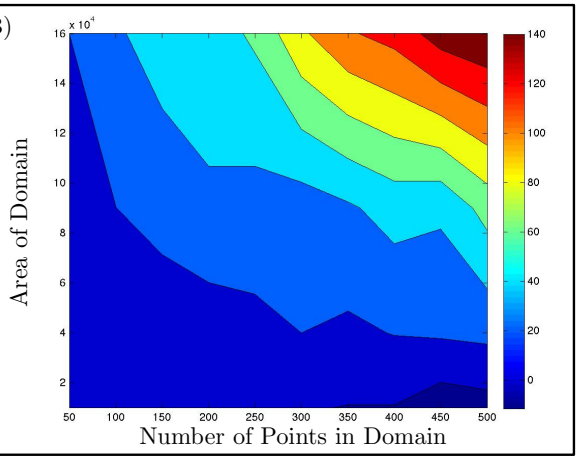

Figure 6. A) Average computation time in seconds of the MILP and the brute force algorithm for problems with variable domain area and variable number of points. B) Contour plot of the difference between the computation time of the brute force algorithm and MILP. A positive value implies that the brute force algorithm took longer to execute than the MILP.

$t_C = 37 + 41 = 78$ where aircraft B pushes back with a right push back maneuver. Using the MILP graph as constraints the optimal schedule is defined as $t_B = 0, t_A = 13$ and $t_C = 13 + 26 = 39$ where aircraft B pushes back with a left push back maneuver. In this scenario the MILP approach provides an increase of throughput of 2 times over the conservative approach. This increase in throughput comes at the cost of smaller push back windows for each aircraft.

V. Computational Performance of the MILP

Here we compare the performance of the MILP with a brute force algorithm for computing the optimal combination of push back sub-windows. The brute force algorithm systematically walks through the green domain searching for feasible combinations of sub-windows.

The brute force algorithm pseudo code is presented in Algorithm 1. We initialize the optimal cost to zero. Given the scheduled spot time of aircraft j, t_j, we know the value of the left and right borders of our green rectangle and store those values as left and right. We then enter a nested for loop where the outer loop goes through the values of t_j^S and the inner loop goes through t_j^F. Once the values for t_j^S and t_j^F have been selected, we can solve for the variables t_i^S and t_i^F that provide us the largest push back interval that is conflict free for the fixed values of t_j^S and t_j^F.

Once we have the four values $(t_i^S, t_i^F, t_j^S, t_j^F)$ we can use the cost function in objective (3) to compute a cost. We then compare this cost to the previously known optimal cost. If the computed cost is greater than the previously known optimal cost we store the computed cost as the known optimal and store the four variables $(t_i^S, t_i^F, t_j^S, t_j^F)$ that define the combination of sub-windows. When the algorithm ends we return the optimal sub-windows that contain no conflicts.

We measure the computation time of the two algorithms where the area of the domain is variable and the number of conflict points is variable. Because we are interested in the computation time we solve a sample problem where the area of the feasible domain is considered to be d^2 for $d = 100s, 200s, .., 500s$. For a fixed domain size, we randomly sample $k = 100, 200, .., 500$ points from the uniform distribution defined within the domain and use both the MILP and brute force algorithm to solve for the optimal combination of push back sub-windows. For a fixed domain size and fixed number of points, we repeat the routine of randomly sampling points from the uniform distribution a total of fifty times and average the computation time. Figure 6A reports the average computation time of the two algorithms on a 1.6 GHz Intel(R) Core(TM) i7 running MATLAB 2011b.

Figure 6B illustrates the difference in the computation time of the brute force algorithm and the MILP. The contour lines are plotted illustrating the gradient in the difference in computation time. A positive value indicates that the MILP executed in less time than the brute force algorithm. The contour plot is color coded where the color red (blue) illustrates where the MILP outperforms the most (least). Figure 6B shows that the brute force only outperforms the MILP for a small subset of problems defined by a small area of domain and a large number of conflict points. As can be seen by the shape of the contour lines, the difference in computation time is affected by changes in both the area of the domain and the number of conflict points.

VI. MILP Computation Time for Solutions of Multiple Aircraft

In section III we formulated a MILP that returns the optimal combination of push back sub-windows for fixed taxiway spot schedules of two aircraft. Here we apply the MILP for fixed taxiway spot schedules of more than two aircraft. We assume that we are given the taxiway spot times for n aircraft, $t_1, t_2, .., t_n$, and the optimal solution is a combination of push back windows for all aircraft $1, 2, .., n$.

The objective function is defined by (28) where the summation is over n aircraft. Constraints that are seen in equations (29)-(36) are generated using Algorithm 2 and constraints that are seen in equations (37)-(41) are generated using Algorithm 3. The number of constraints that are passed to the MILP is a function of the number of conflict points. For every conflict point κ, we generate five constraints that ensure the conflict point is not a convex combination of the optimal push back intervals.

Figure 7 illustrates the computation time on a 1.6 GHz Intel(R) Core(TM) i7 running MATLAB 2011b for taxiway spot schedules of $n = 4, 5, 6$ aircraft. The computation time is shown as a function of the maximum number of pairwise conflict points that we allow within the domain. As can be seen in the figure, the computation time of the MILP significantly increases with the increase in the number of aircraft.

VII. Conclusion and Future Work

In this paper we formulated a MILP to solve for the optimal combination of push back sub-windows. Solutions were constrained to be conflict free in the presence of trajectory uncertainties. The MILP was used to solve for the optimal combination of push back sub-windows for any scheduled spot time difference at a resolution of 1 second.

We analyzed example solutions that illustrate the ability of the MILP to solve for the push back sub-windows regardless of the complexity of the distribution of conflict points. This is critical as the shape and structure of the conflicts is not known *a priori*. Using the computed solutions we generated a minimum time-separation graph for different sequence of aircraft. We compared the MILP minimum-time separation graph to the conservative minimum-time separation graph. We found that the throughput of the conservative schedule can be significantly increased within the ramp area by exploiting the structure of the conflict points.

Now that we understand how to exploit the structure of the conflicts, we would like to integrate the MILP with a state-of-the-art optimal taxiway scheduler. This would allow for the optimal planning of surface operations from the runways all the way to the gate. We would also like to improve upon the computational performance of the MILP. We will also investigate new techniques to reduce the computation

time of the MILP, a critical component for any real time application.

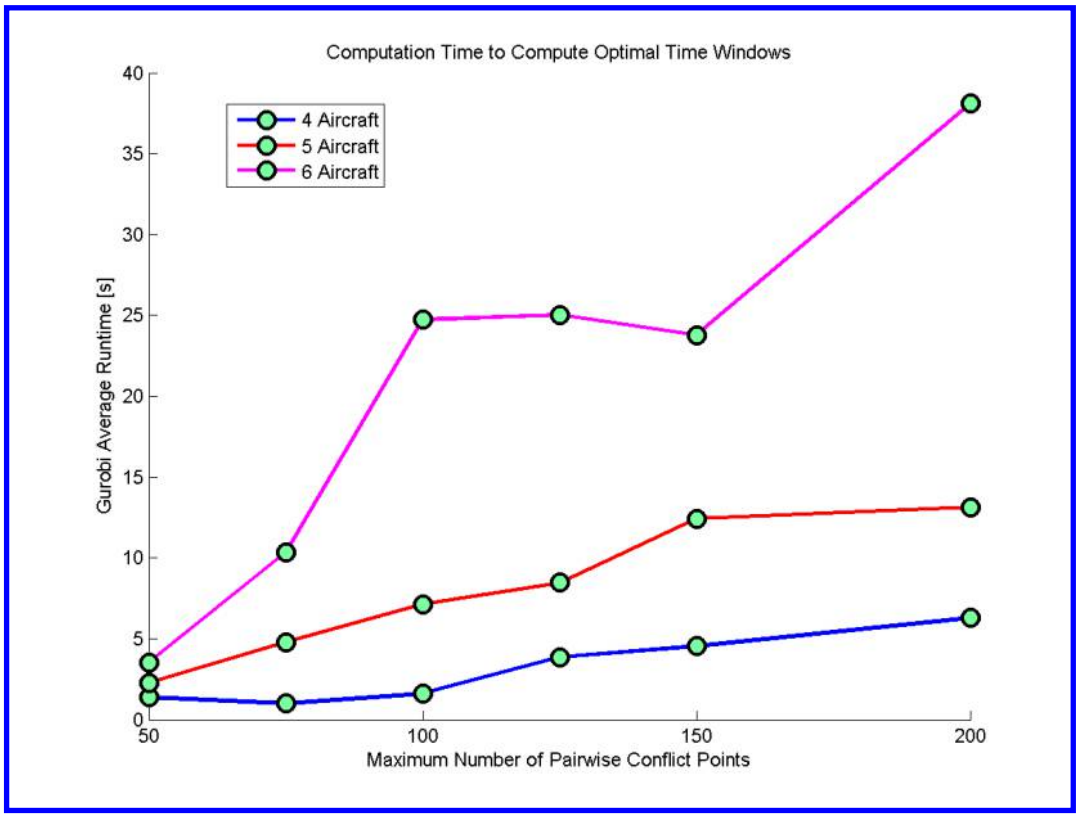

Figure 7. Computation time of solutions for taxiway spot schedules of $n = 4, 5, 6$ aircraft.

Algorithm 2 Multi Aircraft Initial Constraint Generation

For every aircraft i we generate the initial constraints
for i = 1:NumAircraft **do**
$t_i^F - t_i^S - M \geq 0$
$t_i^F - t_i^S - \delta_{min} \geq 0$
$t_i^S - t_i - t_i^{S0} \geq 0$
$t_i^F - t_i - t_i^{F0} \leq 0$
end for

Algorithm 3 Multi Aircraft Conflict Constraint Generation

For every pairwise conflict between aircraft i and j we generate the constraints
for i = 1:NumAircraft -1 **do**
for j = i+1:NumAircraft **do**
for κ =1:NumConflicts **do**
$t_i^F - PB^i - (1 - z_{\kappa 1})S \leq 0$
$t_i^S - PB^i + (1 - z_{\kappa 2})S \geq 0$
$t_j^F - PB^j - (1 - z_{\kappa 3})S \leq 0$
$t_j^S - PB^j + (1 - z_{\kappa 4})S \geq 0$
$z_{\kappa 1} + z_{\kappa 2} + z_{\kappa 3} + z_{\kappa 4} = 1$
end for
end for
end for

References

[1]Gilbo, E. P., "Airport capacity: representation, estimation, optimization," *IEEE Transactions on Control Systems Technology*, Vol. 1, No. 3, 1993, pp. 144–154.

[2]Jung, Y., Hoang, T., Montoya, J., Gupta, G., Malik, W., Tobias, L., and Wang, H., "Performance Evaluation of a Surface Traffic Management Tool for Dallas/Fort Worth International Airport," *9th USA/Europe Air Traffic Management Research and Development Seminar*, Berlin, Germany, 2011.

[3]Pesic, B., Durand, D., and Alliot, J. M., "Aircraft ground traffic optimisation using a genetic algorithm," *Genetic and Evolutionary Computation Conference (GECCO)*, San Francisco, CA, 2001.

[4]Aliot, J. M., Gotteland, J.-P., Durand, N., and Page, E., "Aircraft ground traffic optimization," *4th International Air Traffic management R&D Seminar*, Santa Fe, NM, 2001.

[5]Nemhauser, G. L. and Wolsey, L. A., *Integer and combinatorial optimization*, Wiley-Interscience, New York, NY, USA, 1988.

[6]García, J., Berlanda, A., Molina, J., and Casar, J., "Optimization of airport ground operations integrating genetic and dynamic flow management algorithms," *AI Commun*, Vol. 18, 2005, pp. 143–164.

[7]Herrero, J. G., Berlanda, A., Molina, J. M., and Casar, J. R., "Methods for operations planning in airport decision support systems," *Appl Intell*, Vol. 22, 2005, pp. 183–206.

[8]Balakrishnan, H. and Chandran, B. G., "Algorithms for Scheduling Runway Operations Under Constrained Position Shifting," *Operations Research*, Vol. 58, No. 6, 2010, pp. 1650–1665.

[9]Rathinam, S., Wood, Z., Sridhar, B., and Jung, Y., "A generalized dynamic programming approach for a departure scheduling problem," *AIAA Guidance, Navigation, and Control Conference*, 2009, pp. 10–13.

[10]Smeltink, J. W., Sooner, M. J., de Waal, P. R., and van der Mei, R. D., "An optimisation model for airport taxi scheduling," *INFORMS Annual Meeting*, Denver, CO, 2004.

[11]Marín, A., "Airport management: taxi planning," *Ann Oper Res*, Vol. 143, 2007, pp. 191–202.

[12]Roling, P. C. and Visser, H. G., "Optimal airport surface traffic planning using mixed-integer linear programming," *Int J Aerospace Eng*, Vol. 2008, 2008, pp. 1–12.

[13]Clare, G. L. and Richards, A. G., "Optimization of Taxiway Routing and Runway Scheduling," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 4, 2011, pp. 1000–1013.

[14]Malik, W., Gupta, G., and Jung, Y. C., "Managing departure aircraft release for efficient airport surface operations," *Proceedings of the AIAA Guidance, Navigation, and Control (GNC) Conference*, Toronto, Ontario, Canada, 2010.

[15]Malik, W., Gupta, G., and Jung, Y. C., "Spot Release Planner: Efficient Solution for Detailed Airport Surface Trffic Optimization," *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, Indianapolis, IN, 2012.

[16]Gupta, G., Malik, W., and Jung, Y. C., "A Mixed Integer Linear Program for Airport Departure Scheduling," *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, Hilton Head, SC, 2009.

[17]Gupta, G., Malik, W., and Jung, Y. C., "Incorporating Active Runway Crossings in Airport Departure Scheduling," *AIAA Guidance, Navigation, and Control Conference (GNC)*, Toronto, Ontario, Canada, 2010.

[18]Balakrishnan, H. and Jung, Y., "A framework for coordinated surface operations planning at DFW international airport," *AIAA Guidance, Navigation, and Control Conference (GNC)*, Reston, VA, 2007.

[19]Rathinam, S., Montoya, J., and Jung, Y., "An optimization model for reducing aircraft taxi times at the DFW international airpiort," *26th International Conference of the Aeronautical Sciences*, Anchorage, AK, 2008.

[20]Anderson, R. and Milutinović, D., "An approach to optimization of airport taxiway scheduling and traversal under uncertainty," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 227, No. 2, February 2013, pp. 273–284.

[21]Atkins, S., Brinton, C., and Jung, Y., "Implication of Variability in Airport Surface Operations on 4-D Trajectory Planning," *Proceedings of the 8th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, Anchorage, AK, 2008.

[22]Ravizza, S., Atkin, J. A. D., Maathuis, M. H., and Burke, E. K., "A combined statistical approach and ground movement model for improving taxi time estimations at airports," *JORS*, Vol. 64, No. 9, 2013, pp. 1347–1360.

[23]Gupta, G., Malik, W., and Jung, Y. C., "Effect of Uncertainty on Deterministic Runway Scheduling," *11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, Virginia Beach, VA, 2010.

[24]Lee, H. and Balakrishnan, H., "Optimization of Airport Taxiway Operations at Detroit Metropolitan Airport (DTW)," *In AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, Fort Worth, TX, 2010.

[25]Lee, H. and Balakrishnan, H., "A Comparison of Two Optimization Approaches for Airport Taxiway and Runway Scheduling," *In Digital Avionics Systems Conference*, Williamsburg, VA, 2012.

[26]Coupe, W. J., Milutinović, D., Malik, W., Gupta, G., and Jung, Y., "Robot Experiment Analysis of Airport Ramp Area Time Constraints," *AIAA Guidance, Navigation, and Control Conference (GNC)*, Boston, MA, 2013.

[27]Coupe, W. J., Milutinović, D., Malik, W., Gupta, G., and Jung, Y., "Integration of Uncertain Ramp Area Aircraft Trajectories and Generation of Optimal Taxiway Schedules at Charlotte Douglas (CLT) Airport," *AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, Dallas, TX, 2015.

[28]"Website of Gurobi Optimizer Mathematical Program Solver," http://www.gurobi.com/.